

```
n=2468 # '2468'
```

```
#n=75391
```

```
#n=23456789
```

```
print ('Lambda:')
```

```
if len(str(n))==len(list(filter(lambda x: int(x)%2==0 ,str(n)))):|
```

```
    print(n, 'chứa toàn số chẵn')
```

```
elif len(str(n))==len(list(filter(lambda x: int(x)%2==1 ,str(n)))):
```

```
    print(n, 'chứa toàn số lẻ')
```

```
else:
```

```
    print(n, 'gồm cả số chẵn và số lẻ')
```

```
1 n=2468 # '2468'
2 #n=75391
3 #n=23456789
4 print('Lambda:')
5 if len(str(n))==len(list(filter(lambda x: int(x)%2==0 ,str(n)))):
6     print(n,'chứa toàn số chẵn')
7 elif len(str(n))==len(list(filter(lambda x: int(x)%2==1 ,str(n)))):
8     print(n,'chứa toàn số lẻ')
9 else:
10    print(n, 'gồm cả số chẵn và số lẻ')
```



Python 3.9.5 (tags/v3.9.5:0a7dcdbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)]



Lambda:  
1468 gồm cả số chẵn và số lẻ

In[3]:

```
temp.py x Bai133_Tr53.py x
1 n=2468 # '2468'
2 #n=75391
3 #n=23456789
4 print_('Lambda:')
5 if len(str(n))==len(list(filter(lambda x: int(x)%2==0 ,str(n)))):
6     print(n,'chứa toàn số chẵn')
7 elif len(str(n))==len(list(filter(lambda x: int(x)%2==1 ,str(n)))):
8     print(n,'chứa toàn số lẻ')
9 else:
10    print(n, 'gồm cả số chẵn và số lẻ')
```

```
Bai133_Tr53 x
Python 3.9.5 (tags/v3.9.5:0a7dcdb, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)]
> Lambda:
2468 chứa toàn số chẵn
In[3]: |
```

```
#n=23456789
S=str(n)
print ('Lambda:')
if len(S)==len(list(filter(lambda x: int(x)%2==0 ,S))):
    print(n, 'chứa toàn số chẵn')
elif len(S)==len(list(filter(lambda x: int(x)%2==1 ,S))):
    print(n, 'chứa toàn số lẻ')
else:
    print(n, 'gồm cả số chẵn và số lẻ')
print ('Comprehension:')
if all(int(x)%2==0 for x in S):
    print(n, 'chứa toàn số chẵn')
elif all(int(x)%2==1 for x in S):
    print(n, 'chứa toàn số lẻ')
else:
    print(n, 'gồm cả số chẵn và số lẻ')
```

```
#n=23456789
S=str(n)
print('Lambda:')
if len(S)==len(list(filter(lambda x: int(x)%2==0 ,S))):
    print(n,'chứa toàn số chẵn')
elif len(S)==len(list(filter(lambda x: int(x)%2==1 ,S))):
    print(n,'chứa toàn số lẻ')
else:
    print(n, 'gồm cả số chẵn và số lẻ')

print('Comprehension:')
if all(int(x)%2==0 for x in S):
    print(n, 'chứa toàn số chẵn')
elif all(int(x)%2==1 for x in S):
    print(n, 'chứa toàn số lẻ')
else:
    print(n, 'gồm cả số chẵn và số lẻ')
```

```
1 #Bài tập về nhà: 133+134 trang 53 và từ 117 đến 127 trang 51-52
2 n = int(input('Nhập số nguyên dương n: '))
3 lst = list(str(n))
4 print(lst)
5 lst2 = [y for y in lst if int(y)%2 != 0]
6 print(lst2)
7 if len(lst2)==len(lst):
8     print('Các chữ số toàn lẻ')
9 else:
10    print('Các chữ số không toàn lẻ')
```

if len(lst2)==len(lst)

```
➤ Nhập số nguyên dương n: >? 2468
➤ ['2', '4', '6', '8']
➤ []
➤ các chữ số không toàn lẻ
+ In[3]:
```

134/. Số may mắn: giả sử người ta cho rằng 1 số gọi là số may mắn nếu chỉ chứa toàn các số 6 hoặc số 8. Viết chương trình cho nhập số nguyên n, xét xem n có là số may mắn hay không?

Ví dụ:  $n=686 \Rightarrow 686$  là số may mắn.

$n=68626 \Rightarrow 68626$  KHÔNG phải số may mắn.

```
n=8866668686868

s=str(n)
print('Lambda:')
if len(s)==len(list(filter(lambda x: x=="6" or x=="8",s))):
    print(n,'Là số may mắn')
else:
    print(n,'KHÔNG là số may mắn')

print('Comprehension:')
```

```
if len(s)==len(list(filter(lamb...
```

```
sys.path.extend(['D:\\TTTH_KHTN\\Python270\\Buoi09', 'D:/TTTH_KHTN/Python270/Buoi09'])
```

```
Python 3.9.5 (tags/v3.9.5:0a7dcdb, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)]
```

```
Lambda:
```

```
8866668686868 Là số may mắn
```

```
Comprehension:
```

```
In[3]:
```



```
n=88666687686868
```

```
S=str(n)
```

```
print_('Lambda:')
```

```
if len(S)==len(list(filter(lambda x: x=="6" or x=="8",S))):
```

```
    print(n,'Là số may mắn')
```

```
else:
```

```
    print(n,'KHÔNG là số may mắn')
```

```
print_('Comprehension:')
```

```
sys.path.extend(['D:\\TTTH_KHTN\\Python270\\Buoi09', 'D:/TTTH_KHTN/Python270/Buoi09'])
```



```
Python 3.9.5 (tags/v3.9.5:0a7dcdb, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)]
```

```
Lambda:
```

```
88666687686868 KHÔNG là số may mắn
```

```
Comprehension:
```

```
In[3]:
```

```

if len(S)==len(list(filter(lambda x: x=="6" or x=="8",S))):
    print(n, 'Là số may mắn')
else:
    print(n, 'KHÔNG là số may mắn')

```

cách 1

```

print('Comprehension 1:') # "8686"
lst=[x for x in S if x=="6" or x=="8"]
print('List=', lst)
if len(S)==len(lst):
    print(n, 'Là số may mắn')
else:
    print(n, 'KHÔNG là số may mắn')

```

cách 2

```

print('Comprehension 2:') # "8686"
if all(x=="6" or x=="8" for x in S):
    print(n, 'Là số may mắn')
else:
    print(n, 'KHÔNG là số may mắn')

```

cách 3

```
n=86866
```

```
S=str(n)
```

```
print('Lambda:')
```

```
if len(S)==len(list(filter(lambda x: x=="6" or
```

```
    print(n, 'Là số may mắn')
```

```
else:
```

```
    print(n, 'KHÔNG là số may mắn')
```

Python 3.9.5 (tags/v3.9.5:0a7dcbb, May 3 2021,

Lambda:

86866 Là số may mắn

Comprehension 1:

List= ['8', '6', '8', '6', '6']

86866 Là số may mắn

Comprehension 2:

86866 Là số may mắn

In[3]: |



# THAO TÁC TRÊN TẬP TIN & THƯ MỤC



Lê Văn Hạnh  
levanhhanhvn@gmail.com

## 1.1. Mở file

- Trước khi muốn đọc hoặc ghi file, cần có thao tác mở file.
- Cú pháp:

**`fileObject = open(fileName [, accessMode] [, buffering] [, encoding='utf-8'])`**

trong đó:

- ***fileName***: tên file sẽ truy cập (kèm đường dẫn nếu file không có trong thư mục hiện hành).
- ***buffering***: gồm các giá trị:
  - ***1*** : có sử dụng buffer,
  - ***0*** : không sử dụng buffer,
  - ***>1*** : buffer size,
  - ***<0*** : default size. Đây là giá trị mặc định.
- ***encoding='utf-8'***: sử dụng khi mở file unicode.



## 1.1. Mở file

*xem tiếp lý thuyết trong tài liệu hardcopy của thầy*

– Cú pháp:

**fileObject = open(fileName [, accessMode] [, buffering] [, encoding='utf-8'])**

trong đó:

- **accessMode**: chế độ mở tập tin: read, write, append, ... Có một số chế độ mở file là:
  - **r** : mở để đọc nội dung (mặc định)
  - **w** : mở để ghi nội dung
  - **a** : mở để ghi thêm nội dung vào cuối file.
  - **r+** : mở để đọc và ghi. Con trỏ nằm ở đầu file.
  - **w+** : mở để đọc và ghi. Ghi đè nếu file đã tồn tại, nếu file chưa tồn tại thì tạo file mới để ghi.
  - **a+** : mở để đọc và thêm vào cuối file. Con trỏ nằm ở cuối file. Nếu file chưa tồn tại thì tạo file mới để ghi.

Mặc định là mở file *text*, nếu muốn mở file dạng nhị phân (*binary*) thì thêm **b**, như: **rb**, **wb**, **ab**, **rb+**, **wb+**, **ab+**.

```
1 import csv
2 def DocNoiDungFile_CSV_1(filename):
3     try:
4         with open(filename, 'rt') as f:
5             csvReaderObj = csv.reader(f, delimiter=',')
6             temp = csvReaderObj
7             print("In toàn bộ kết quả đọc từ file")
8             print(list(temp))
9             |
10            print("Tổng cộng gồm %d dòng" % csvReaderObj.line_num)
11 except FileNotFoundError:
12     print("Không tìm thấy tập tin %s" % filename)
13 #-----
14 DocNoiDungFile_CSV_1('D:\\ProgrammingLanguage.csv')
```

*thay tất cả bằng 2 dấu \\ hoặc bằng /*

```
"C:\Program Files\Python39\python.exe" "C:\Program Files\JetBrains\PyCharm Community I
```

```
import sys; print('Python %s on %s' % (sys.version, sys.platform))  
sys.path.extend(['D:\\TTTH_KHTN\\Python270\\Buoi09', 'D:/TTTH_KHTN/Python270/Buoi09'])
```

```
Python 3.9.5 (tags/v3.9.5:0a7dcdb, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)]:
```

In toàn bộ kết quả đọc từ file

```
[['Programming language', ' Designed by', ' Appeared', ' Extension'], ['Python', ' Gu:
```

Tổng cộng gồm 5 dòng

In[3]:



15/.Viết chương trình thực hiện các yêu cầu sau, trong đó, mỗi yêu cầu được viết riêng thành hàm. Sau đó chương trình chính sẽ gọi các hàm này thực hiện:

a. Hàm đọc file *Departments.csv* và in ra màn hình theo mẫu sau đây, trong đó:

- Tên tiêu đề cột được in canh giữa 2 ký tự '|' ở 2 đầu
- Field dạng chuỗi được in canh trái
- Field dạng số được in canh phải.

Minh họa kết quả thực hiện:

DEPARTMENTS LIST			
DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing	114	1700
...			
...			
270	Payroll		1700

Tổng cộng gồm 27 dòng dữ liệu

- Ví dụ 1: viết hàm sử dụng *csv.reader* đọc file *ProgrammingLanguage.csv* ở trên, in tất cả kết quả đọc được theo từng dòng ra màn hình:

```
import csv
def DocNoiDungFile_CSV_1(filename):
    try:
        with open(filename, 'rt', encoding='utf8') as f:
            csvReaderObj = csv.reader(f, delimiter=',')
            for row in csvReaderObj:
                print(row)
            print("Tổng cộng gồm %d dòng" % csvReaderObj.line_num)
    except FileNotFoundError:
        print("Không tìm thấy tập tin %s" % filename)
#-----

DocNoiDungFile_CSV_1('D:\\ProgrammingLanguage.csv')
```

tham  
khảo  
code  
này để  
làm bài  
tập trên

```
1 import csv
2 def DocNoiDungFile_CSV_1(filename):
3     try:
4         with open(filename,'rt')as f:
5             csvReaderObj = csv.reader(f, delimiter=',')
6             for row in csvReaderObj:
7                 print(row)
8             print("Tổng cộng gồm %d dòng" %csvReaderObj.line_num)
9     except FileNotFoundError:
10         print("Không tìm thấy tập tin %s" % filename)
11     #-----
12
13 DocNoiDungFile_CSV_1('D:\\Departments.csv')
```

```
"C:\Program Files\Python39\python.exe" "C:\Program Files\JetBrains\PyCharm Community I

import sys; print('Python %s on %s' % (sys.version, sys.platform))
sys.path.extend(['D:\\TTTH_KHTN\\Python270\\Buoi09', 'D:/TTTH_KHTN/Python270/Buoi09'])

Python 3.9.5 (tags/v3.9.5:0a7dcbbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)]
['department_id', 'department_name', 'manager_id', 'location_id']
['10', 'Administration', '200', '1700']
['20', 'Marketing', '201', '1800']
['30', 'Purchasing', '114', '1700']
['40', 'Human Resources', '203', '2400']
['50', 'Shipping', '121', '1500']
In[3]:
```

```
3 try:
4     with open(filename, 'rb', encoding='utf8') as f: #read text/binary
5         csvReaderObj = csv.reader(f)
6         # Lấy dòng đầu tiên chứa các tên field ra đưa vào 1 list
7         lst=csvReaderObj.__next__()
8         #Sau khi lệnh trên thực hiện, biến lst có dạng
9         #['Programming language', 'Designed by', 'Appeared', 'Extension']
10        #In tiêu đề cột
11        print('|',lst[0].center(20),'|', lst[1].center(20), '|', lst[2].center(10), '|', lst[3].center(13),'|')
12        print('_' * 75)
13        for row in csvReaderObj:
14            print('{: <21s}'.format(row[0]),'|', '{: <20s}'.format(row[1]), '| %10s | %13s | ' % (row[2], row[3]))
15            print(f'Dữ liệu gồm {csvReaderObj.line_num-1} dòng')
16    except FileNotFoundError:
17        print("Không tìm thấy tập tin %s" % filename)
18    #----- I
19    DocNoiDungFile_CSV_4('D:\\ProgrammingLanguage.csv')
```

## 4.5. Set

- Trong *Python*, *set* là class đại diện cho khái niệm toán học của 1 tập hợp, nghĩa là set gồm tập hợp các phần tử duy nhất (không trùng lặp) và có thể thực hiện các phép toán về tập hợp như: hợp, giao, ....  
*tham khảo thêm trong sách*
- Đặc điểm của *set*:
  - Không giới hạn số lượng phần tử.
  - Có thể thêm hoặc xóa phần tử.
  - Có thể chứa biến thuộc nhiều kiểu dữ liệu khác nhau, nhưng không thể chứa phần tử có thể thay đổi được như *list*, *set* hay *dictionary*.
  - Được tối ưu hóa trong việc kiểm tra xem một phần tử cụ thể có được chứa trong tập hợp hay không nhờ dựa trên cấu trúc dữ liệu bảng băm (*hash table*).

### 4.5.1. Khai báo và gán giá trị cho set

- Tạo bằng phép gán: *set* được tạo bằng cách đặt tất cả các phần tử trong dấu ngoặc nhọn `{}` và phân tách nhau bằng dấu phẩy `(,)`.
- Tạo set bằng cách sử dụng hàm *set()*.

```
1 def NhapSet():
2     myset=set()
3     while True:
4         x=int(input("Nhập số để thêm vào set (nhập -1 để kết thúc):"))
5         if x== -1:
6             return myset
7         else:
8             myset.add(x)
9
10 #===== chương trình chính =====
11 set01=NhapSet()
12 print(set01)
```



Python 3.9.5 (tags/v3.9.5:0a7dcdb, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)]

Nhập số để thêm vào set (nhập -1 để kết thúc):>? 0

Nhập số để thêm vào set (nhập -1 để kết thúc):>? 5

Nhập số để thêm vào set (nhập -1 để kết thúc):>? 9

Nhập số để thêm vào set (nhập -1 để kết thúc):>? 7

Nhập số để thêm vào set (nhập -1 để kết thúc):>? 3

Nhập số để thêm vào set (nhập -1 để kết thúc):>? 45

Nhập số để thêm vào set (nhập -1 để kết thúc):>? -1

{0, 3, 5, 7, 9, 45}

In[3]:



68/. Tổ chức chương trình dưới dạng các hàm chức năng để thực hiện việc xử lý trên set như sau:

Yêu cầu:

- Viết hàm cho phép người dùng lần lượt nhập các phần tử số cho *set* cho đến khi giá trị nhận vào là -1. Hàm trả về *set* vừa nhập. Sử dụng hàm này để tạo ra 2 *set*: *set1* và *set2*.
- In các số có trong *set1* và *set2* ra màn hình.
- Cho biết mỗi *set* có bao nhiêu phần tử, tổng giá trị các phần tử của mỗi *set*.
- Tìm giá trị lớn nhất, nhỏ nhất của mỗi *set*.
- Tìm những số có trong *set1* hoặc *set2*.
- Tìm những số có trong cả 2 *set* *set1* và *set2*.
- Tìm những số có trong *set1* nhưng không có trong *set2*.
- Tìm những số có trong *set1* hoặc *set2* nhưng số đó không được có trong cả 2 *set*.
- Sắp xếp *set1* tăng dần và *set2* giảm dần.

```
1 def NhapSet(stt):
2     myset=set()
3     while True:
4         x=int(input("Nhập số để thêm vào set "+str(stt)+" (nhập -1 để kết thúc):"))
5         if x== -1:
6             return myset
7         else:
8             myset.add(x)
9
10    #===== chương trình chính =====
11    set01=NhapSet(1)
12    print(set01)
13    set02=NhapSet(2)
14    print(set02)
```

## 3.5. Phối hợp các đối tượng dạng sequence type

### 3.5.1. List & Set

70/. Viết chương trình tạo 1 list L gồm n phần tử có giá trị ngẫu nhiên từ 0 đến 10, với n do người dùng nhập. Tính tổng các số trong list sau khi đã loại bỏ các số trùng nhau (chỉ giữ lại 1 số trong các số trùng nhau).

Ví dụ:  $L = [1, 2, 2, 2, 3, 5, 5] \Rightarrow \text{tổng} = 11$



73/. Viết chương trình cho nhập số điện thoại (S). In ra các số từ 0 đến 9 không xuất hiện trong số điện thoại vừa nhập.

Ví dụ: nhập `S='0913158020'`

⇒ Trong số điện thoại 0913158020 không chứa các ký số: [4, 6, 7]

👉 Gợi ý:

- Tạo set1 chứa tất cả các ký số từ 0-9. Tương tự bài tập trước, cũng có 2 cách để tạo set:
  - Cách 1: tự khai báo 1 set trong đó liệt kê đầy đủ các ký số từ 0-9
  - Cách 2: sử dụng thuộc tính `string.digits` trong module `string`.
- Tạo set2 chứa tất cả các ký số có trong số điện thoại.
- Sử dụng phép bù giữa set1 và set2 để tìm kết quả

```
temp.py x Bai70_Tr43.py x Bai73_Tr44.py x Bai15_Tr57.py x Bai68_Tr42.py x Bai121_Tr51.py x Bai133_Tr53.py x
D:\TTTH_K 1 s=input('Nhập số điện thoại:')
Bai15_Tr57.py 2 NumberSet=set('0123456789')
Bai68_Tr42.py 3 print(NumberSet)
Bai70_Tr43.py 4
Bai73_Tr44.py 5 for i in NumberSet:
Bai121_Tr51.py 6     if i not in s:
Bai133_Tr53.py 7         print(i,end=' ')
temp.py 8
```

for i in NumberSet > if i not in s

Bai133\_Tr53 x temp x Bai134\_Tr53 x Bai121\_Tr51 x Bai15\_Tr57 x Bai68\_Tr42 x Bai68\_Tr42 (1) x Bai70\_Tr43 x Bai73\_Tr44 x

```
import sys; print('Python %s on %s' % (sys.version, sys.platform))
sys.path.extend(['D:\\TTTH_KHTN\\Python270\\Buoi09', 'D:/TTTH_KHTN/Python270/Buoi09'])

Python 3.9.5 (tags/v3.9.5:0a7dcdb, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)]
Nhập số điện thoại:>? 123456
{'4', '9', '7', '2', '1', '6', '0', '8', '3', '5'}
9 7 0 8
In[3]:
```

72/. Cho người dùng nhập 1 chuỗi (S). Cho biết S có chứa đầy đủ tất cả các ký tự từ A-Z hay không? Không phân biệt ký tự hoa/thường.

👉 Gợi ý:

- Tạo ra 1 set chứa tất cả các ký tự thường từ a-z bằng 1 trong 2 cách sau:
  - Cách 1: tự khai báo 1 set trong đó liệt kê đầy đủ các ký tự từ a-z
  - Cách 2: sử dụng thuộc tính `ascii_lowercase` trong module `string`.
- Chuyển tất cả các ký tự trong S thành ký tự thường.
- Sử dụng 1 trong các toán tử so sánh (`>`, `>=`, `<`, `<=`, `==`, ...) để tập hợp vừa có, nếu bằng nhau sẽ cho ra `True`, ngược lại sẽ cho `False`.
- Sử dụng 2 chuỗi sau để kiểm tra:
  - *The quick brown fox jumps over the lazy dog* (đủ các ký tự từ a-z)
  - *The quick brown fox jumps over the lazy cat* (thiếu 2 ký tự d và g)

```
1  chuoiNhap = input("Moi nhap chuoi: ")
2  chuoiNhap=set(chuoiNhap.lower())
3  print(chuoiNhap)
4  chuThuong = []
5  for i in range (97,123):
6      chuThuong.append(chr(i))
7  set(chuThuong)
8
9  print("Cac chu ko co : ",end='')
10 for i in chuThuong:
11     if i not in chuoiNhap:
12         print(i,end='')
```



```
temp.py x Bai72_Tr43.py x Bai70_Tr43.py x Bai73_Tr44.py x Bai15_Tr57.py x Bai68_Tr42.py x Bai121_Tr51.py x
Buoi09 D:\TTTH_K 1 s=input('Nhập số điện thoại:')
2 NumberSet=set({'0','1','2','3','4','5','6','7','8','9'})
3 print(NumberSet)
4 sdt=set(s)
5 print(sdt)
6 print("Các số không có trong sdt vừa nhập là:", NumberSet-sdt)
7
8
```

```
Bai133_Tr53 x temp x Bai134_Tr53 x Bai121_Tr51 x Bai15_Tr57 x Bai68_Tr42 x Bai68_Tr42 (1) x Bai70_Tr43 x Bai73_Tr44 x
sys.path.extend(['D:\\TTTH_KHTN\\Python270\\Buoi09', 'D:/TTTH_KHTN/Python270/Buoi09'])
Python 3.9.5 (tags/v3.9.5:0a7dcbbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)]
Nhập số điện thoại:>> 456321
{'3', '2', '8', '4', '0', '1', '5', '7', '9', '6'}
{'3', '2', '5', '4', '1', '6'}
{'0', '9', '8', '7'}

In[3]:
```

```
1 chuoINHap = input("Moi nhap chuoI: ")
2 chuoINHap=set(chuoINHap.lower())
3 print(chuoINHap)
4 chuThuong = []
5 for i in range (97,123):
6     chuThuong.append(chr(i))
7 set(chuThuong)
8
9 print("Cac chu ko co : ",chuThuong-chuoINHap)
```

4.4.	Dictionary .....	105
4.4.1.	Giới thiệu.....	105
4.4.2.	Tạo dictionary .....	106
4.4.3.	Thêm phần tử vào dictionary (hoặc cập nhật value thông qua key).....	108
4.4.4.	Truy xuất phần tử của dictionary .....	108
4.4.5.	Kiểm tra một key đã có trong dictionary hay chưa? .....	109
4.4.6.	Chuyển đổi list sang dictionary .....	109
4.4.7.	Chuyển đổi 2 thành phần key và value cho nhau.....	110
4.4.8.	Xóa trên dictionary .....	110
4.4.9.	Gộp 2 dictionaries .....	111
4.4.10.	Sắp xếp dictionary .....	112

## 4.4. Dictionary

### 4.4.1. Giới thiệu

- Dictionary cũng là một cấu trúc dạng *sequence*. Dictionary gồm nhiều phần tử không có thứ tự, mỗi phần tử sẽ là 1 bộ **key:value** (cấu trúc Dictionary tương tự như một *object json*). Trong đó:
  - *key*: giá trị của *key* là duy nhất (không trùng) và sau này không thể chỉnh sửa. Kiểu dữ liệu có thể là *number*, *string*, *tuple*.
  - *value*: có giá trị tùy ý. Do đó có thể chỉnh sửa/cập nhật sau đó.
- Dictionary thường được sử dụng khi chúng ta có một số lượng lớn dữ liệu. Các *dictionary* được tối ưu hóa để trích xuất dữ liệu với điều kiện phải biết được *key* để lấy giá trị.
- Khác biệt giữa Dictionary và List, Tuple: List, Tuple sử dụng *index* để phân biệt các phần tử, còn Dictionary thì dùng các *key* để phân biệt.

## 3.3. Dictionary

### 3.3.1. Xử lý trên dictionary

47/. Cho người dùng nhập 1 chuỗi (S).

- Đếm số lượng ký số, số lượng ký tự có trong S.
- Tính tổng các ký số có trong S.

Ví dụ: `S= 'Python 3.9.1'  $\Rightarrow$  13`

```

temp.py x Bai72_Tr43.py x Bai47_Tr36.py x Bai70_Tr43.py x Bai73_Tr44.py x Bai15_Tr57.py x Bai68_Tr42.py x Bai121_Tr51
Buoi09 DATTH_K
Bai15_Tr57.py
Bai47_Tr36.py
Bai68_Tr42.py
Bai70_Tr43.py
Bai72_Tr43.py
Bai73_Tr44.py
Bai121_Tr51.py
Bai133_Tr53.py
Bai134_Tr53.py
temp.py
External Libraries
Scratches and Con
1 S=input("Nhập chuỗi: ")
2 print("Câu A:")
3 mydict={"kytu":0, "kyso":0}
4 for c in S:
5     if c.isdigit():
6         mydict["kyso"]+=1
7     elif c.isalpha():
8         mydict["kytu"]+=1
9 print("Chuỗi %s gồm %d ký tự và %d ký số" %(S,mydict["kytu"],mydict["kyso"]))

```

for c in S &gt; elif c.isalpha()

```

Bai133_Tr53 x temp x Bai134_Tr53 x Bai121_Tr51 x Bai15_Tr57 x Bai68_Tr42 x Bai68_Tr42 (1) x Bai70_Tr43 x Bai73_Tr44 x Bai47_Tr36 x
Python 3.9.5 (tags/v3.9.5:0a7dcdbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)]
> Nhập chuỗi: >? sài gòn 12
> Câu A:
> Chuỗi sài gòn 12 gồm 6 ký tự và 2 ký số
In[3]:

```

```

S = {str}
c = {str}
mydict = {}
Special

```

```
temp.py x Bai72_Tr43.py x Bai47_Tr36.py x Bai70_Tr43.py x Bai73_Tr44.py x Bai15_Tr57.py x Bai68_Tr42.py x Bai121_Tr57
Buoi09 D:\TTTH_K
Bai15_Tr57.py
Bai47_Tr36.py
Bai68_Tr42.py
Bai70_Tr43.py
Bai72_Tr43.py
Bai73_Tr44.py
Bai121_Tr51.py
Bai133_Tr53.py
Bai134_Tr53.py
temp.py
External Libraries
Scratches and Con
4
5     if c.isdigit():
6         mydict["kyso"]+=1
7     elif c.isalpha():
8         mydict["kytu"]+=1
9     print("Chuỗi %s gồm %d ký tự và %d ký số" %(s,mydict["kytu"],mydict["kyso"]))
10
11     print("Câu B:")
12     tong=0
13     for c in S:
14         if c.isdigit():
15             tong=tong+int(c)
16     print("Tổng các số có trong s là:", tong)
```

Bai133\_Tr53 x temp x Bai134\_Tr53 x Bai121\_Tr51 x Bai15\_Tr57 x Bai68\_Tr42 x Bai68\_Tr42 (1) x Bai70\_Tr43 x Bai73\_Tr44 x Bai47\_Tr36 x

Chuỗi sai gon quận 12 gồm 10 ký tự và 2 ký số  
Câu B:  
Tổng các số có trong s là: 3  
In[5]:

S = {str}  
c = {str}  
mydict = {dict}  
tong = {int}  
Special